ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"SISTEMA PARA EL ANÁLISIS Y PROCESAMIENTO DE LOS LOGS DE LOS SERVIDORES DE RED DE LA FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN (FIEC) DE LA ESPOL USANDO HADOOP"

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO EN COMPUTACIÓN
ESPECIALIZACION SISTEMAS TECNOLÓGICOS

INGENIERO EN COMPUTACIÓN
ESPECIALIZACION SISTEMAS INFORMACIÓN

Presentada por:

JOSUÉ JEFFERSON GUARTATANGA ROBAYO

EDDY ROBERTO ESPINOSA DAQUILEMA

Guayaquil - Ecuador
2010

AGRADECIMIENTO

A Dios, que nos ha conservado con vida y salud.
A nuestros padres, quienes han sido y son un pilar fundamental en nuestras vidas.
A nuestros profesores, por todos los conocimientos y consejos transmitidos

DEDICATORIA

A Dios
A nuestros padres
A nuestros familiares
A nuestros amigos

TRIBUNAL DE SUSTENTACIÓN

PROFESORA DE LA MATERIA DE GRADUACIÓN

MsC. Carmen Vaca R.

PROFESOR DELEGADO POR EL DECANO

Ing. Juan Moreno V.

DECLARACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Josué Jefferson Guartatanga Robayo

Eddy Roberto Espinosa Daquilema

RESUMEN

El propósito de este proyecto es implementar un sistema que permita el análisis y procesamiento de los logs de los servidores de red del laboratorio de la Facultad de Ingeniería en Electricidad y Computación (FIEC) de la ESPOL.

Estos logs son de gran tamaño por lo que resulta indispensable utilizar una plataforma de procesamiento masivo y escalable de datos, para ello se ha usado el framework Hadoop.

El documento está dividido en 6 capítulos que comprenden el planteamiento del problema, el marco teórico, el análisis del problema con su respectivo diseño de la solución, la implementación y análisis de los resultados, las pruebas realizadas y posteriormente las conclusiones y recomendaciones.

En el primer capítulo se define la problemática actual, se indican los objetivos y justificación del proyecto, así como también el alcance del mismo.

En el segundo capítulo se expone el marco teórico utilizado para este trabajo, se explican conceptos como: la plataforma Hadoop, el modelo MapReduce, Hive como una herramienta de consultas y los servicios web ofertados por Amazon usados para las pruebas.

El tercer capítulo explica el diseño de la solución del problema. En este capítulo se establecen las responsabilidades que tendrá cada uno de los

componentes en los que ha sido dividido el proyecto, así como los datos de entrada y salida que se manejarán en cada etapa.

En el cuarto capítulo se explican los detalles considerados en el código usado para la implementación y el funcionamiento del sistema. Se enfatiza también en el manejo de los diferentes formatos encontrados en los archivos de los logs procesados.

El quinto capítulo describe las pruebas realizadas utilizando los servicios web de Amazon (Amazon Web Services). Como resultado de las pruebas se incluyen gráficos sobre el tiempo requerido para procesar los datos utilizando diferentes cantidades de nodos.

Finalmente en el sexto capítulo se presentan las conclusiones obtenidas, y se plantean recomendaciones para futuros trabajos relacionados.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En los últimos 15 años se ha visto un crecimiento acelerado de los servicios en la web, tanto así que existen actualmente más de 200 millones de sitios publicados; todos estos alojados en cientos de millones de servidores (56% Apache, 25% Microsoft) (1).

Muchos de estos servidores mantienen información sumamente importante, convirtiéndose en un blanco sensible para spammers o crackers; por lo que las herramientas que provean información que permitan identificar accesos irregulares, serán de gran importancia para los administradores de redes.

Todos y cada uno de los eventos que ocurran en un servidor durante el funcionamiento normal de los servicios que brinda, son registrados en archivos denominados logs. El estudio de los datos que contienen estos logs, puede contribuir con información relevante que ayude a los administradores de redes a tomar decisiones.

Datos importantes a considerar en el análisis de un log son por ejemplo: las operaciones que han fallado o que se han realizado correctamente, en qué día y a qué hora el usuario navega, qué protocolo y puerto está usando, qué navegador es el más utilizado (2), etc.

La idea de implementar el proyecto se planteó como respuesta a los requerimientos del Ing. Juan Moreno, administrador de redes de la FIEC.

Estos requerimientos consisten básicamente en el procesamiento y depuración de los logs de los servidores de la Facultad.

Una vez refinada esta información, será más fácil para el administrador determinar si hubo o no algún spam, considerando la frecuencia de acceso que han tenido los usuarios, así como los correos electrónicos masivos.

Debido a que todas las operaciones realizadas sobre un servidor se van registrando en los logs, el tamaño de estos archivos puede crecer de forma indiscriminada. Es por esto que se decidió utilizar una plataforma que permita procesamiento masivo y distribuido para el desarrollo del sistema.

CAPÍTULO 1

1. PLANTEAMIENTO

1.1 Definición del problema

Cada servidor, dependiendo de su implementación y/o configuración, podrá o no generar información acerca de todas las actividades que se realicen sobre éste.

Esta información es almacenada en logs (archivos de texto plano que puede llegar a tener millones de registros), y para ser aprovechada, necesita una depuración previa facilitando el análisis de su contenido.

Cada uno de los archivos generados por sus respectivos servidores, mantienen un formato diferente para almacenar los datos y el proceso de identificar los campos principales puede volverse complejo.

Debido a que todo programa necesita información consistente, es necesario realizar un pre-procesamiento que genere archivos con un formato estándar haciendo manejable la información.

Se podrá determinar el nivel de efectividad de los controles de seguridad que se tengan implantados: firewall, antivirus-spam-filtrado de contenidos URL, sistema de detección y prevención de intrusiones (3).

Actualmente tomaría varios días el procesamiento manual de los archivos de logs con la información de eventos registrados en un mes, lo cual hace indispensable implementar una herramienta que automatice el proceso de extracción y consolidación de la información.

1.2 Objetivos

El presente trabajo tiene como objetivo general aprovechar el conjunto de datos que tienen los logs generados en los servidores del laboratorio de la facultad, a fin de presentar información relevante que pueda ser analizada por los administradores del mismo.

Se planificaron los siguientes objetivos específicos en el contexto del objetivo principal/general:

Determinar los campos que se extraerán de cada uno de los logs que se generan en los servidores Samba, Maillog y Apache Tomcat (HTTP); para así puntualizar cómo manipularlos.

Implementar una aplicación distribuida para procesar datos extraídos de logs de servidores de red.

Diseñar una interfaz gráfica que permita mostrar información importante que respalde la toma de decisiones en tiempo oportuno.

1.3 Justificación

Actualmente existen buenas herramientas que analizan logs ya implementadas. Sin embargo, estas herramientas presentan algunas desventajas frente a la aplicación propuesta. Algunas de estas desventajas se enumeran a continuación:

Requieren el pago de una licencia (algunos bordean los 200 USD como el Analizador de Registros Absoluto versión Pro cuyo precio es de 250 USD).

No corren en plataforma distribuida.

Detectan automáticamente los formatos de los archivos de registros pero no muestran los resultados deseados, por ejemplo:

Muestran el número de visitantes por día pero no indica cuáles; lo mismo sucede con los hits.

Son rápidos al procesar archivos pequeños pero presentan deficiencias en cuanto al rendimiento cuando el tamaño de los archivos excede unos cuantos gigas (M5 Analyzer posee dificultad al procesar logs mayores a 10Gb).

Las herramientas que permiten analizar logs presentan desventajas considerables frente a la aplicación distribuida que se desarrollará usando Hadoop. La limitada capacidad que se admite para subir los

archivos justifica el porqué del desarrollo de nuestra herramienta; algunos programas cargan los datos a ser analizados en memoria, y por lo tanto no podría hacerse efectivo el análisis de éstos cuando el tamaño de la información está en el orden de los Gigabytes.

Con el desarrollo de un sistema que maneja un mecanismo distribuido de procesamiento de datos, lo que se busca es optimizar el tiempo de tratamiento usando procesamiento paralelo.

1.4 Alcance

Este proyecto básicamente proveerá información referente a los siguientes servidores: Cedro, Ceibo y Palma.

La información a obtener con respecto al servidor Cedro será la siguiente:

Visitas a Cedro; las visitas a los recursos/páginas, ya sea desde el interior o exterior de la ESPOL. Esta información puede ser filtrada por protocolo, mes e ítem (cantidad de registros del 1-20). Al consultar los datos se mostrará una tabla con las siguientes columnas: ítem, fecha, IP, protocolo y frecuencia; todos ordenados descendentemente por frecuencia.

Recursos de Cedro; el recurso como tal, al que se accedió, para lo cual se mostrará la fecha del recurso visitado con su respectiva

frecuencia (cantidad de visitas del 1-20). La consulta de los recursos podrá ser filtrada por mes y por ítem. Al consultar los datos se mostrará una tabla con las siguientes columnas: número de ítem, fecha, recurso y frecuencia; ordenados descendentemente por frecuencia.

Navegadores de Cedro; los hits de cada navegador, filtrados ya sea por mes o por ítem (cantidad de visitas 1-20). La consulta mostrará una tabla con las siguientes columnas: número de ítem, fecha, navegador y frecuencia ordenados descendentemente por frecuencia.

La información a obtener con respecto al servidor Ceibo será la siguiente:

Los correos que con más frecuencia envían e-mails. La información puede ser filtrada por: remitente/ destinatario; por protocolo (HTTP, HTTPS, HFS, SSH, ICMP, DNS, IMAP, SMTP, ESMTP o todos); por mes (uno en especial o todos) y también por número de ítems (cantidad de registros que se quiere visualizar, puede ir del 1-20).

La información a obtener con respecto al servidor Palma será la siguiente:

Accesos a Palma: Los usuarios de las personas que más visiten el servidor Palma. Esta información puede ser filtrada por ítem (cantidad

de registros del 1-20) y fecha (mes). La consulta mostrará una tabla con las siguientes columnas: ítem, fecha, usuario y frecuencia.

Recursos de Palma: Los recursos que son usados en la visita a las páginas. Esta información puede ser filtrada por ítem (cantidad de registros del 1-20) y fecha (mes). La consulta mostrará una tabla con las siguientes columnas: ítem, fecha, recurso y frecuencia.

CAPÍTULO 2

2. MARCO TEÓRICO

El objetivo del estudio de este capítulo es abarcar de manera general los fundamentos teóricos para la comprensión del funcionamiento de la plataforma Hadoop.

2.1 Map/Reduce

Map/Reduce es un modelo desarrollado por Google para la implementación de aplicaciones que utilizan computación paralela y que manejan grandes cantidades de datos.

Este paradigma plantea que toda aplicación paralela pueda ser escrita utilizando 2 fases: la fase de **Map** y la fase R**educe**. La primera fase producirá datos estructurados y la segunda fase los consumirá.

La fase Map transforma los datos de entrada en datos intermedios; extrayendo una clave y asociando a esta clave un valor que dependerá del tratamiento que se le quiera dar a los datos.

Estos datos no necesitan ser necesariamente del mismo tipo de los de ingreso, por lo que las entradas pueden mapear a cero o a varias salidas.

El Combine es un post mapeado, el cual toma los valores intermedios, los asocia a una clave común y a continuación son agrupados obteniendo una lista por clave que será enviada a la fase Reduce.

La fase Reduce recibe las listas generadas *(**clave;** valor)* por el Mapper, procesa la información y genera el resultado final dependiendo del control que se especifique (4).

En la Figura 2-1 se grafica el esquema principal de Map/Reduce.

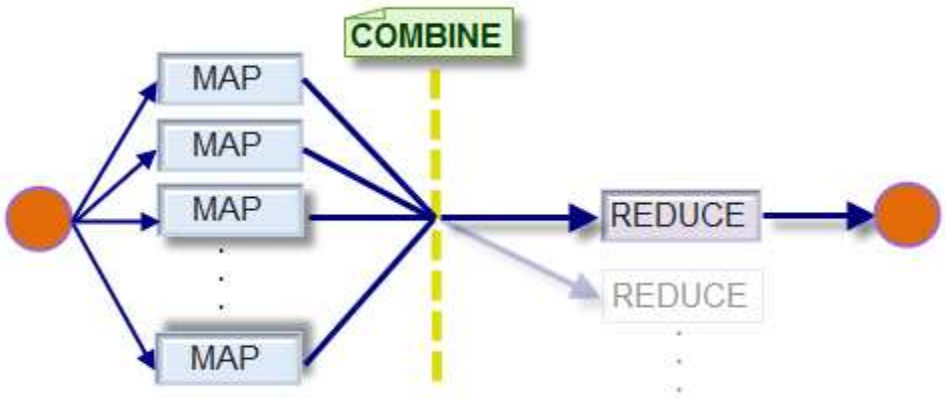


Figura 2-1Esquema Map Reduce

2.2 Hadoop

Hadoop es una plataforma que nos permite desarrollar aplicaciones que necesitan escalabilidad, ya que puede manipular y procesar grandes cantidades de información (petabytes) (5).

Una de las más importantes ventajas de Hadoop es que manipula los detalles de procesamiento, permitiéndole al desarrollador enfocarse sólo en la lógica del negocio.

Es posible correr Hadoop en un cluster de servidores ya que cuenta con un sistema de archivos distribuido (HDFS) repartiendo la carga de datos entre los nodos. Al ubicar la información de forma distribuida, la búsqueda se puede hacer rápidamente pudiendo acceder a ella de forma paralela.

Tanto el paradigma Map/Reduce como el HDFS fueron diseñados en un entorno que soporta fallos (6), manteniendo un nivel de seguridad para recuperación y realización de un backup automático. Y si por algún motivo se presentara un error en un nodo, éste no afectará el resultado puesto que se maneja redundancia de nodos.

En la Figura 2-2 se muestra un diagrama de las fases manejadas por la plataforma Hadoop.

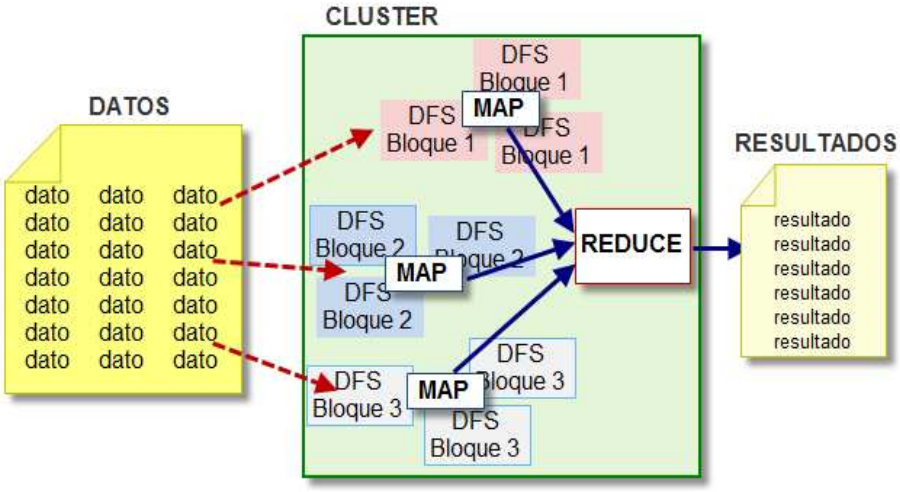


Figura 2-2 Plataforma Hadoop

2.3 Hive

Es una infraestructura para data warehouse que provee sumarización de datos y soporta Ad-hoc queries.

Sus aplicaciones están en:

- Procesamiento de Logs
- Minería de datos
- Indexación de Documentos
- Inteligencia de negocios (ejemplo Google Analytics)
- Modelamiento predictivo y prueba de hipótesis

Está expresada en un lenguaje declarativo parecido al SQL denominado Hive QL que soporta múltiples inserciones y es muy extensible permitiendo manipular scripts, que ayudan a personalizar los querys para optimizar las búsquedas.

Pueden contener cualquier tipo de datos: primitivos (Integers, flotantes, Strings, Dates y Booleanos); colecciones de datos (arreglos y mapas); así como combinaciones de los mismos.

Hive incluye un sistema de catálogos que contiene la metadata especificada durante la creación de las tablas y reusada cada vez que éstas sean referenciadas; y ayuda en la exploración de datos para mejorar el rendimiento de la consulta (7).

2.4 Servicios Web de Amazon

Los servicios Web de Amazon (AWS) son un conjunto de servicios informáticos ofrecidos por Amazon a través de Internet, los cuales son controlados de forma remota.

De entre los servicios más importantes brindados por Amazon se tienen:

Elastic Compute Cloud (EC2), permite a los usuarios alquilar ordenadores virtuales en los que se pueden ejecutar aplicaciones con un alto nivel de redundancia (8).

Simple Storage Service (S3), Facilita el almacenamiento de archivos de gran tamaño; permitiendo levantar clústeres con un número determinado de procesadores, todo bajo demanda (9).

CAPÍTULO 3

3. ANÁLISIS Y DISEÑO

3.1 Análisis del problema

Cuando se analizó el problema como tal se descubrió una aplicación que podía ser ampliamente aprovechada por la Facultad para facilitar el tratamiento de los logs de los diferentes servidores que se usan: Cedro, Ceibo y Palma.

Los tres problemas relevantes son:

1. Actualmente la información almacenada en los logs no es utilizada para analizar accesos no regulares a los servidores. Por lo tanto la información de estos archivos utiliza espacio en disco pero no provee una mayor utilidad para el administrador.

2. No es posible escribir un solo mapper para procesar archivos, puesto que cada programa servidor genera logs con formato diferente. Será necesario escribir programas que parseen la información tomando en cuenta las particularidades de cada tipo de log a procesar. Palma es un caso especial que se analizará pues los datos, referentes a un registro en particular, no son contiguos. Algunos registros contienen un carácter de

salto de línea, lo que ocasionaría que al utilizar procesamiento distribuido los datos de un registro se envíen a dos nodos distintos.

3. La cantidad de información almacenada en una archivo log, para un mes, puede representar varios gigas. Procesar esta cantidad de datos, para obtener información relevante que nos permita tomar medidas para mantener seguros los servidores significaría invertir una cantidad significativa de tiempo, lo cual a su vez representa mayores gastos.

Para resolver el problema mencionado en el punto 1 es importante considerar en el análisis cómo van a condensarse los datos para mostrar información relevante al usuario una vez que los archivos han sido procesados.

El problema mencionado en el punto 2 es uno de los factores esenciales por los cuales la aplicación a implementar utilizará Hadoop, plataforma ideal para procesar archivos de texto de gran tamaño con cierto formato, aunque éste sea complejo.

Para resolver el problema mencionado en el punto 3 se tomará en cuenta los servicios que ofrece Amazon: S3 y EC2, haciendo que la aplicación a implementar sea escalable. Con estos servicios se

ahorrará espacio en disco de los servidores y tiempo de procesamiento.

En la sección siguiente se presenta el diseño de la aplicación a implementar teniendo en cuenta los factores mencionados previamente.

3.2 Diseño de la solución

Al analizar el problema se determinó que se utilizarán 4 pasos: mapeo, reducción, consulta y presentación. En cada una de estas etapas se procesarán los datos como se describe a continuación:

3.2.1 Mapeo

Para cada servidor se determinará los datos a extraer (valor) asociándolos a un identificador (clave).

En esta sección se utilizará como convención el siguiente formato para describir la salida de los Mappers: (***clave*****;** *valor*)

Puede darse el caso que la clave sea compuesta, entonces éstos datos se concatenarán con el carácter de espacio; lo mismo puede suceder con el valor.

La Figura 3-1representa como está dividida la etapa de mapeo.

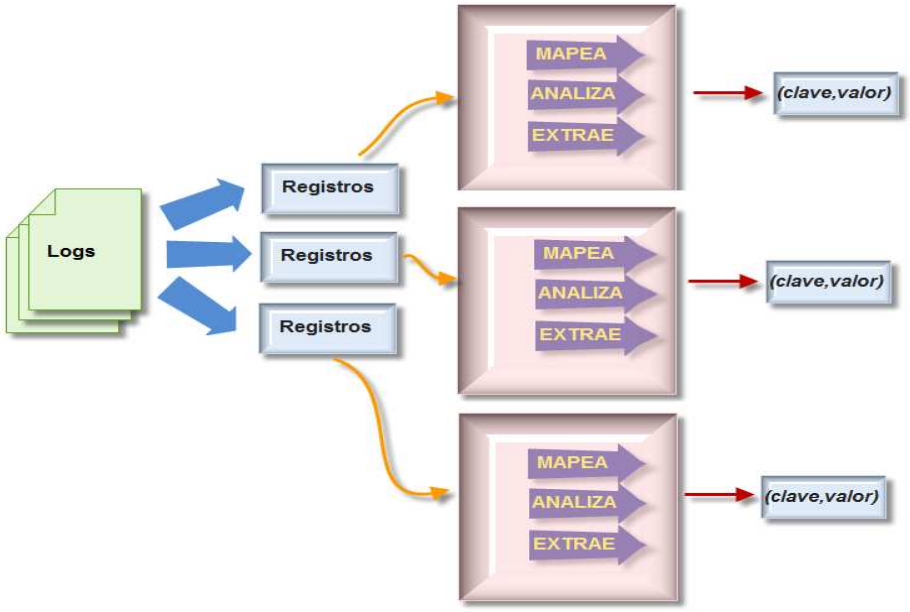


Figura 3-1 Etapa de Mapeo y sus fases

Debido a que cada archivo maneja un formato distinto se implementarán 5 mappers diferentes. En los párrafos siguientes se analizarán las características de cada formato encontrado y la información a extraer.

Cedro:

La entrada del mapper de Cedro son registros de los logs HTTP; de éstos se extraerá: la IP del visitante, la fecha de registro y el protocolo por el cual se accedió. Estos datos formarán la clave, a la cual se asociará el entero 1 como valor, logrando identificar una visita de

alguien específico en una determinada fecha y para un protocolo específico. Se sumarán después (no en el mapper), todos los unos. Entonces la salida de un mapper de Cedro será *(**IP, fecha, protocolo**; uno)*.

Ceibo:

La entrada del mapper de Ceibo son registros de los archivos maillogs; de éstos se extraerá: el identificador del e-mail, que formará la clave, a la cual se asociarán como valores: la IP del remitente, la fecha de envío del e-mail, la dirección e-mail del remitente/destinatario y el protocolo, logrando identificar el flujo de correos.

El mapper tendrá 2 opciones como salidas, ambas con la misma clave pero con valores distintos; esto se debe a que el protocolo no se encontraba en la misma línea de registro de los maillogs. Entonces la salida de un mapper de Ceibo será:

Opción 1: *(**id e-mail**; IP remitente, fecha, e-mail remitente, e-mail destinatario)*.

Opción 2: *(**id mail**; protocolo)*.

Palma:

La entrada del mapper de Palma_concatena son registros de los logs de Samba, de éstos se extraerá: el nombre del archivo y el índice de la

línea de registro, que formarán la clave, a la cual se asociarán como valores: la primera y segunda parte de un registro no contigüo, que deberá ser unificado posteriormente.

Este mapper tendrá 2 salidas por cada registro encontrado en el log, ambas con la misma clave pero con valores distintos; esto se debe a que el servidor escribe cada registro utilizando dos líneas. Entonces la salida de un mapper de Palma_contatena será:

Opción 1: *(**nombre archivo, índice línea registro**; línea1)*.

Opción 2: *(**nombre archivo, índice línea registro**; línea2)*.

Una vez organizado el log se procederá con el análisis de los accesos y recursos:

La entrada del mapper de Palma_accesos son registros de los logs de Samba concatenados, generados por la salida del reducer de Palma_concatena; de los cuales se extraerá: el usuario que accedió y la fecha de registro para formar la clave, a la cual se asociará el entero 1 como valor, logrando identificar el acceso de alguien específico en una determinada fecha. Se sumarán después (no en el mapper) todos los unos. Entonces la salida de un mapper de Palma_accesos será (***usuario, fecha**; uno*).

La entrada de los mappers de Palma_recursos son registros de los logs de Samba concatenados, generados por la salida del reducer de Palma_concatena; de éstos se extraerá: el recurso que fue accedido y la fecha de registro, que formarán la clave, a la cual se asociará el entero 1 como valor, logrando identificar los recursos más solicitados en una determinada fecha. Se sumarán después (no en el mapper) todos los unos. Entonces la salida de un mapper de Palma_recursos será *(**recurso, fecha**; uno)*.

3.2.2 Reducción

En esta fase los datos de entrada son las salidas de los mappers. Estos se distribuyen, se ordenan y luego se consolidan agrupándolos por clave. Se necesitarán reducers intermedios como combiners.

Las salidas de los reducers tendrán la misma estructura que la de los mappers, pero con valores y claves nuevas. Cabe recalcar que estas salidas se guardarán en archivos de texto.

La Figura 3-2 representa como está dividida la etapa de reducción.

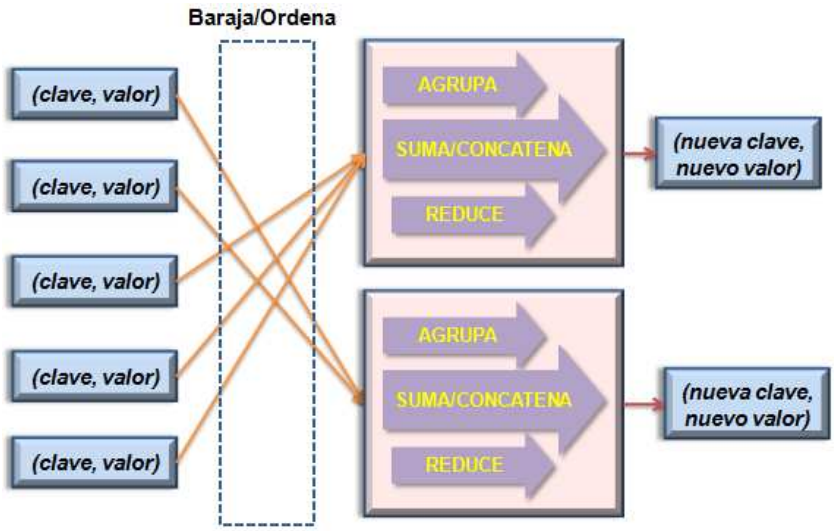


Figura 3-2 Etapa de Reducción y sus fases

Cedro:

Este agrupa todos los valores por clave, obteniendo una secuencia de unos; luego se suman para saber cuántas veces alguien ha accedido desde una cierta IP dentro de un intervalo de fechas y utilizando un protocolo determinado. Entonces la salida del reducer de Cedro será *(**IP, fecha, protocolo**; n)*, donde el entero n representa la sumatoria de todos los unos en la lista.

Ceibo:

Este agrupa todos los valores por clave, asociando la IP remitente, fecha, e-mail remitente, e-mail destinatario con el protocolo

correspondiente al e-mail. La salida del reducer de Ceibo será *(**IP remitente, fecha, e-mail remitente, e-mail destinatario**; protocolo)*.

Palma:

Palma_concatena agrupa todos los valores por clave, asociando las líneas de un solo registro que luego se concatenarán. La salida del reducer de Palma_concatena será *(**nombre archivo, índice línea registro**; líneas concatenadas)*.

Palma_accesos agrupa todos los valores por clave, obteniendo una secuencia de unos; luego se suman para saber cuántas veces alguien ha accedido dentro de un intervalo de fechas. La salida del reducer de Palma_accesos será *(**usuario, fecha**; n)*, donde el entero n representa la sumatoria de todos los unos en la lista.

Palma_recursos agrupa todos los valores por clave, obteniendo una secuencia de unos; luego se suman para determinar cuáles son los recursos más accedidos en un intervalo de tiempo. La salida del reducer de Palma_ recursos será *(**recurso, fecha**; n)*, donde el entero n representa la sumatoria de todos los unos en la lista.

3.2.3 Consultas usando Hive

La entrada en la fase Hive está formada por las salidas de los reducer de Cedro, Ceibo, Palma.

Para cada una de ellas se crearán tablas y se insertarán los registros respectivos.

El uso de Hive permitirá que, una vez procesados un grupo de logs, puedan realizarse consultas sobre esos resultados sin tener que ejecutar nuevamente las etapas de mapeo y reducción.

La Figura 3-3 representa cómo está estructurada la etapa de presentación.

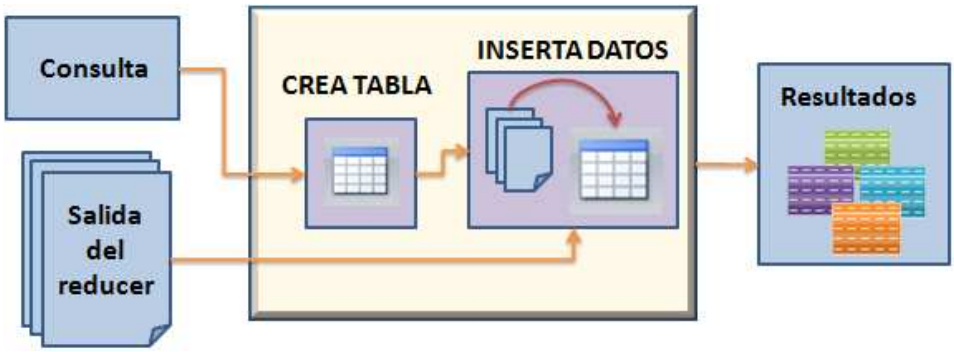
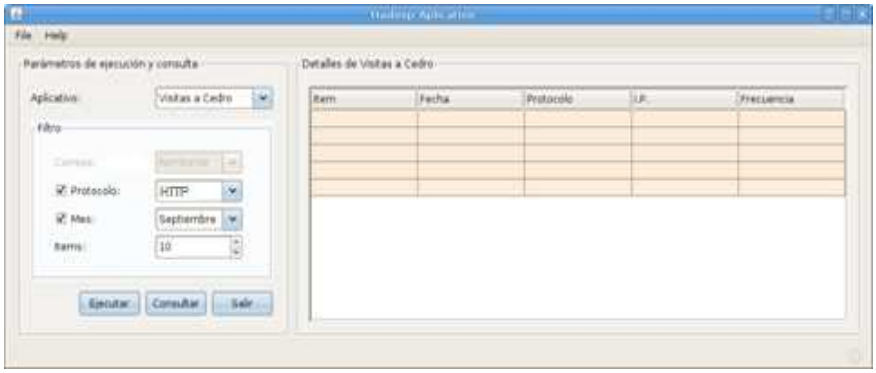


Figura 3-3 Etapa de Consultas Hive y sus fases

3.2.4 Presentación

Para el diseño de la interface gráfica de escritorio se utilizará el IDE de Java Eclipse 3.4.2 con la librearía Swing. La información generada por Hive se presentará en dicha interface cuyo prototipo se muestra La Figura 3-4.



La Figura 3-4 Pantalla Principal de la Aplicación

En síntesis el siguiente gráfico detalla el flujo de información en la aplicación y cómo interactúa cada una de sus fases, desde el origen de los datos (logs) hasta la interfaz de usuario donde se presentarán los resultados.

La Figura 3-5 muestra el todo el proceso de la aplicación.

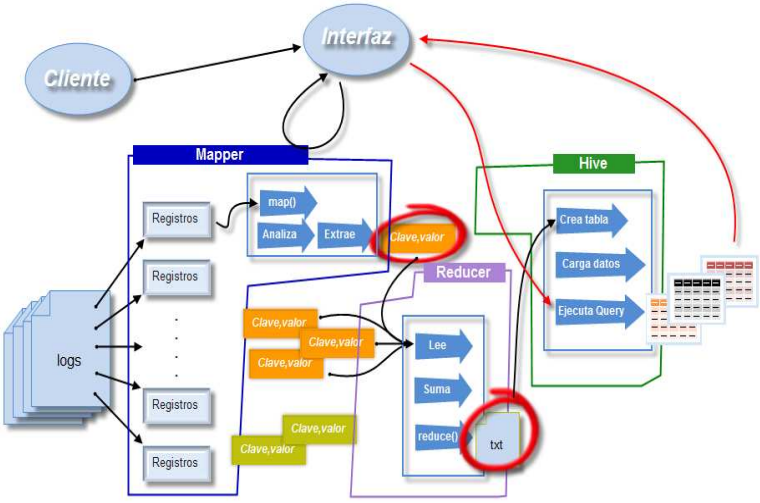


Figura 3-5 Flujo de Datos a través del Diseño de la Solución

CAPÍTULO 4

4. IMPLEMENTACIÓN

En este capítulo se detallará el desarrollo de la implementación, la cual ha sido dividida en:

Extracción y manipulación de los archivos logs.

Operaciones realizadas con el mapper y reducer.

El proceso de presentación de los resultados, en base a consultas tipo SQL, utilizando Hive.

Cabe recalcar que los archivos logs pueden registrar más campos en un registro, de acuerdo a cómo se haya configurado el servidor.

A continuación se analizarán los formatos utilizados en los servidores de FIEC al momento de desarrollar esta aplicación:

4.1 Extracción y manipulación de los archivos logs

Para la extracción de datos de los logs es necesario conocer cuáles son los formatos de éstos.

Cedro posee un Servidor Apache HTTP. El formato de los registros del log es como se describe a continuación:

```
%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""
```

Donde la información a extraer es:

%h, dirección IP del cliente

%t, tiempo que se recibió la solicitud

\"%r\", solicitud del cliente donde se encuentra el recurso que se desea con el protocolo que se utiliza

\"%{User-agent}i\", identificación del navegador

Ejemplo:

67.195.112.238 - - [22/Nov/2009:04:13:29 -0500] "GET /resources/materias/licred/FIEC06221_implementacion_soporte_xp_prof.pdf HTTP/1.0" 200 51590 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp/3.0; HTTP://help.yahoo.com/help/us/ysearch/slurp)"

Ceibo genera archivos tipo maillog. El formato de los registros del log es como se describe a continuación:

Ejemplo:

Nov 9 00:21:16 ceibo milter-greylist: nA95LGs1020166: addr mail.periodicourbano.com[200.125.135.132] from <nobody@mail.periodicourbano.com> to <amera@fiec.espol.edu.ec> delayed for 00:05:00 (ACL 300)

Donde la información a extraer es:

Nov 9, Fecha en que se envió el e-mail

[200.125.135.132], Dirección IP del remitente

<nobody@mail.periodicourbano.com>, E-Mail del remitente

<amera@fiec.espol.edu.ec>, E-mail del destinatario

Palma posee un servidor Samba. El formato de los registros del log es como se describe a continuación:

Ejemplo:

```
[2009/09/16 12:46:56, 1] smbd/service.c:make_connection_snum(1033)
  wrks126-170fiec (200.9.176.170) connect to service apincay initially as user apincay (uid=6828, gid=501) (pid 5491)
[2009/09/17 13:52:46, 2] smbd/open.c:open_file(391) opreciad opened file Internet Download Manager/IDMan.exe read=Yes write=No (numopen=11)
```

Donde la información a extraer es:

[2009/09/16], Fecha de conexión

apincay, El usuario que se ha conectado al servidor

2009/09/17, Fecha de acceso a un recurso del servidor

Internet Download Manager/IDMan.exe, Recurso.

4.2 Implementación de Mappers y Reducers

Para dar facilidad en el soporte del sistema se ha dividido el código fuente en paquetes, representados con nombres de los servidores, los cuales especifican qué se ha implementado en cada uno de ellos.

La Figura 4-1 muestra como está dividido del código del proyecto por paquetes.

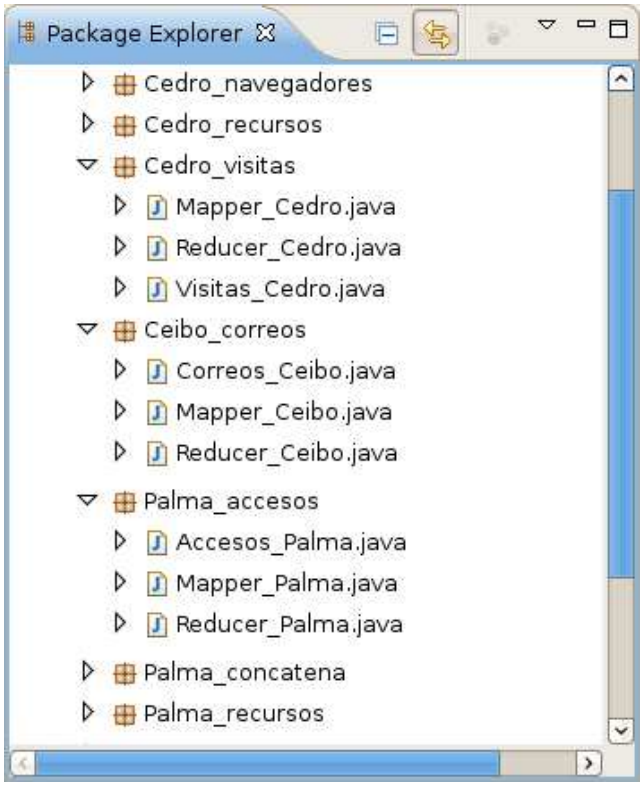


Figura 4-1 Clases por Paquetes

Exceptuando el paquete Palma_concatena, en cada mapper se encuentra la manera de extraer la información de interés, utilizando librerías de manipulación de Strings del API de java.

Como se menciono anteriormente, obviando a los mappers Palma_concatena y Ceibo_correos, todos los demás utilizan el mismo método de extracción de datos. En la sección siguiente se explicará la implementación del mapper de Cedro, dicho esquema puede aplicarse a 3 mappers de los 5 implementados.

4.2.1 Cedro visitas

En la Figura 4-2, se identifica una visita realizada al servidor Cedro permitiendo conocer: “Cuándo”, “quién” y “cómo” se hizo ésta.

Se utiliza una variable de tipo StringBuilder (fec_ip_prot) para almacenar los valores a extraer; éstos serán concatenados utilizando el carácter de tabulación. Los campos que contendrá la variable fec_ip_prot son:

- Fecha, localizada a continuación del primer corchete.
- IP, localizada desde inicio del registro hasta el primer carácter en blanco.
- Protocolo, contenido entre comillas dobles después de un espacio en blanco.

Mapper_Cedro.java | Reducer_Cedro.java | Visitas_Cedro.java

```
public class Mapper_Cedro extends MapReduceBase
    implements Mapper<LongWritable, Text, Text, IntWritable> {
  private final static IntWritable one = new IntWritable(1);
 public Mapper_Cedro() { }

 public void map(LongWritable key, Text value,
   OutputCollector<Text,IntWritable> output,
   Reporter reporter) throws IOException {
     StringTokenizer linea =new StringTokenizer(value.toString(),"\n");
     int aux;
     String protocolo=null;
   while (linea.hasMoreTokens()) {
     String registro = linea.nextToken();
     StringBuilder fec_ip_prot=new StringBuilder();
     aux=0;
     aux=registro.indexOf('[')+1;
     fec_ip_prot.append(registro.substring(aux,aux+11));
     fec_ip_prot.append('\t');
     fec_ip_prot.append(registro.substring(0,registro.indexOf(' ')));
     fec_ip_prot.append('\t');
     aux=registro.indexOf('"')+1;
     protocolo=registro.substring(aux,registro.indexOf('"', aux));
     if (protocolo.split(" ").length==3){
         fec_ip_prot.append(protocolo.split(" ")[2]);
         output.collect(new Text(fec_ip_prot.toString()),one);
     }
   }
 }
}
```

Figura 4-2 Código del Mapper de Cedro

La Figura 4-3 presenta gráficamente la salida del Mapper de Cedro, representados con colores que hacen referencia a la Figura 4-2.

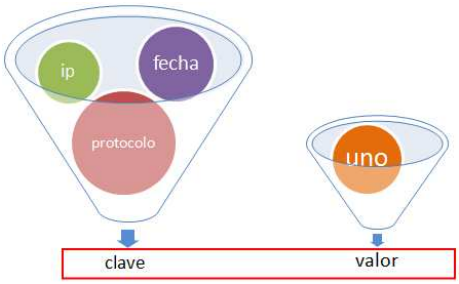
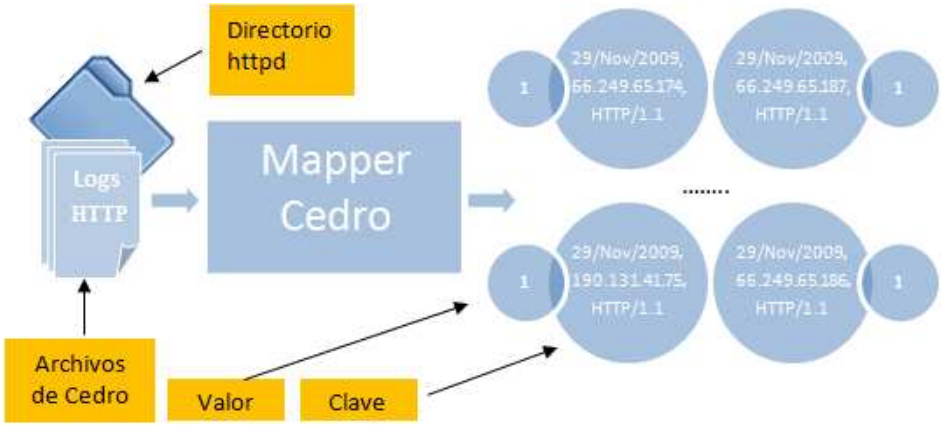


Figura 4-3 Estructura (clave; valor) del Mapper Cedro

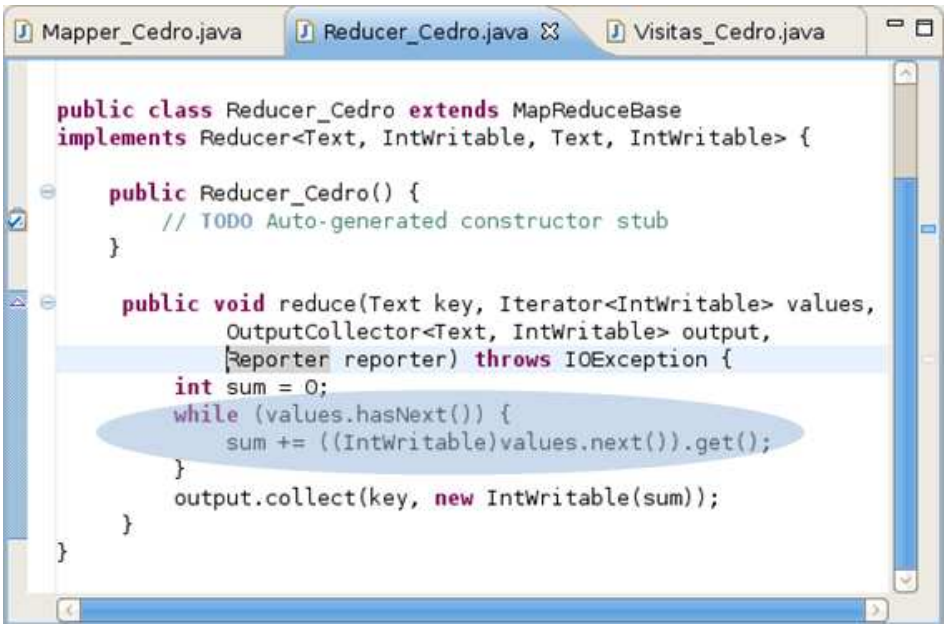
La Figura 4-4 muestra el esquema de interacción del Mapper de Cedro.



La Figura 4-4 Diagrama de Bloques del Mapper de Cedro

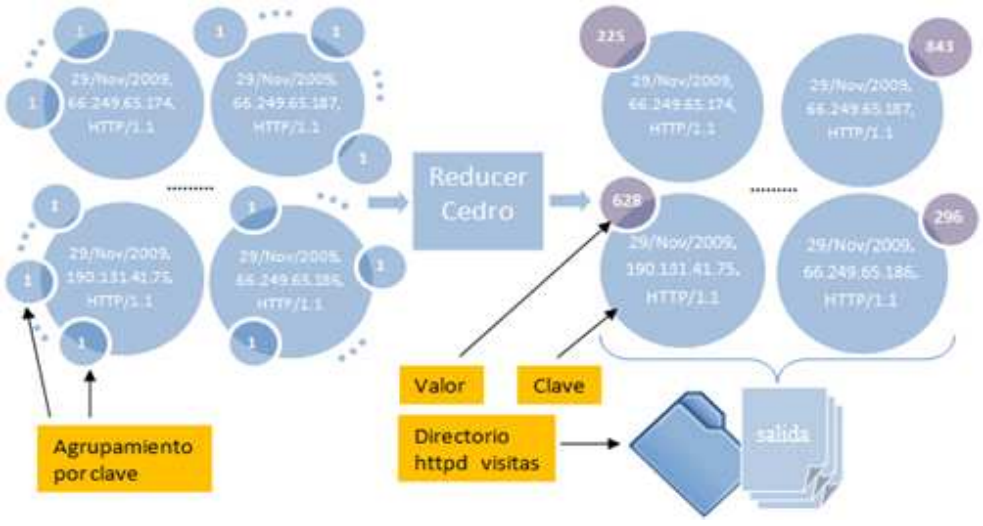
Los reducers: Cedro (navegadores, recursos y visitas), Palma (accesos, Palma_recursos) siguen el modelo de la Figura 4-5 donde se registran los valores que reciben de parte de sus mappers respectivos, y se agrupan por clave, sumando los unos para luego ser almacenados.

La Figura 4-5 bosqueja el modelo para el reducer de Cedro.



La Figura 4-5 Modelo de Reducer de Cedro

La Figura 4-6 muestra el diagrama de bloques del Reducer de Cedro.



La Figura 4-6 Diagrama de Bloques del Reduccer de Cedro

De la salida del reducer se tendrá archivos planos con el siguiente formato:

```
29/Nov/2009   66.249.65.174   HTTP/1.1   225
29/Nov/2009   66.249.65.187   HTTP/1.1   843
…             …               …          …
29/Nov/2009   66.249.65.186   HTTP/1.1   296
```

Estos archivos de texto plano serán utilizados para llenar tablas que serán consultadas por Hive para mostrar el resumen de la información en la interface gráfica.

4.2.2 Ceibo

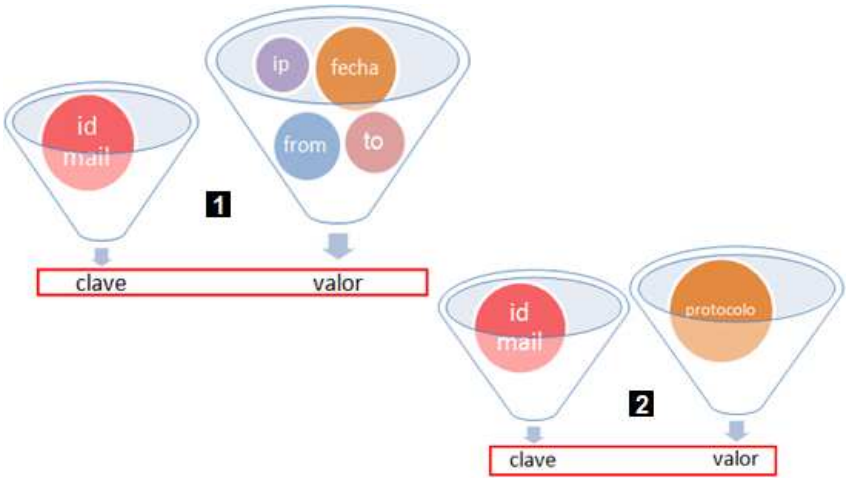
En la implementación, para Ceibo, lo que se busca es asociar el e-mail con su respectivo protocolo. Las etiquetas 1 y 2 de la Figura 4-7 nos indican las opciones de salida del mapper.

Como se observa en la Figura 4-7, se utilizan dos variables de tipo StringBuilder (fec_ip_from_to y Protocolo) para almacenar los valores a extraer como se hizo para el mapper de Cedro. Por esta razón tenemos dos opciones de salida que dependerá de localizar la cadena " from <" por cada registro encontrado. Si existe entonces el valor de la salida será la variable fec_ip_from_to caso contrario Protocolo.

Se usaron colores que asocian y esquematizan cada una de las partes de la salida del mapper de Ceibo.

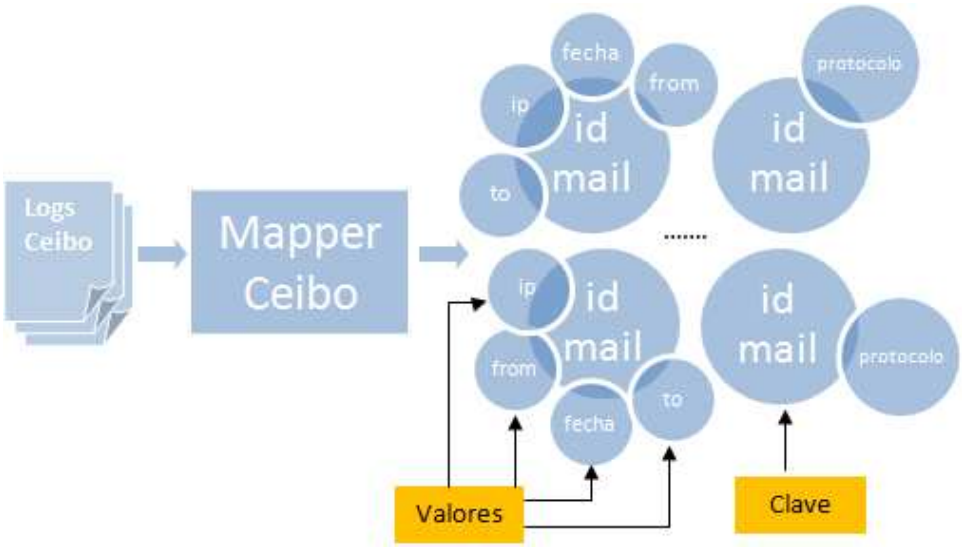
La Figura 4-7 Código Mapper de Ceibo

La Figura 4-8 presenta gráficamente la salida del Mapper de Ceibo, representados con etiquetas y colores que hacen referencia a la Figura 4-7.



La Figura 4-8 Estructura (clave; valor) para el mapper de Ceibo

El Figura 4-9 muestra el diagrama de bloques del Mapper de Ceibo.



La Figura 4-9 Diagrama de Bloques del Mapper de Ceibo

En el reducer se identificó el contenido de la variable *elemento* que puede ser: el *protocolo* o la concatenación de: *fecha, ip, from* y *to* (provenientes del mapper de cedro).

En la Figura 4-10 se diferencia el contenido por cada registro encontrado, que dependiendo de su longitud se almacenan en variables diferentes.

Mapper_Ceibo.java | Reducer_Ceibo.java | Correos_Ceibo.java

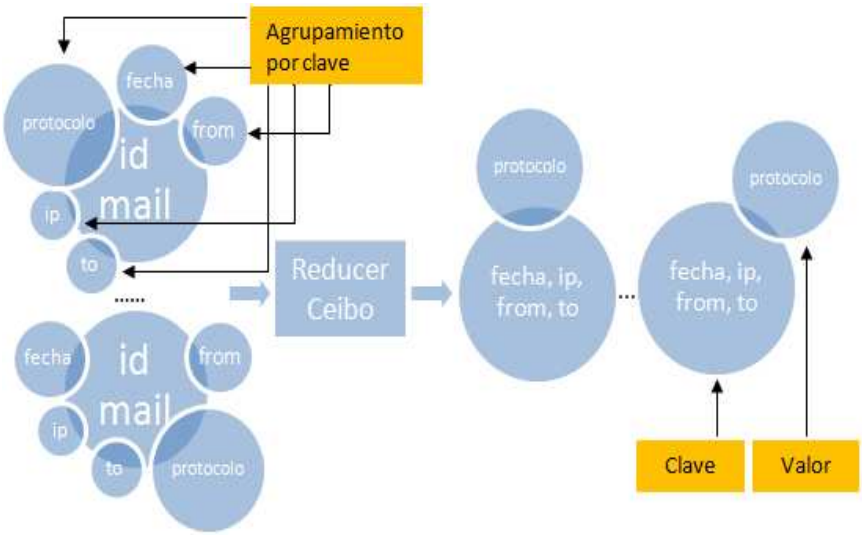
```
public class Reducer_Ceibo extends MapReduceBase
implements Reducer<Text, Text, Text, Text> {
    public Reducer_Ceibo() {}

     public void reduce(Text key, Iterator<Text> values,
             OutputCollector<Text, Text> output, Reporter reporter)
            throws IOException {
        StringBuilder protocolo=new StringBuilder();
         ArrayList<Text> iterator = new ArrayList<Text>();
         while (values.hasNext()){
             StringBuilder elemento=new StringBuilder();
             elemento.append(values.next().toString());
             if (elemento.length()>10)
                 iterator.add(new Text(elemento.toString()));
             else
                 protocolo=elemento;
         }
         for(int i=0;i<iterator.size()-1;i++)
             output.collect(iterator.get(i),new Text(protocolo.toString()));
     }
}
```

La Figura 4-10 Código Reducer de Ceibo

El reducer de Ceibo nos proporciona archivos planos con el formato: fecha, ip, from, to, protocolo.

La Figura 4-11 muestra el diagrama de bloques del Reduccer de Ceibo.



La Figura 4-11 Diagrama de Bloques del Reducer de Ceibo

4.2.3 Palma

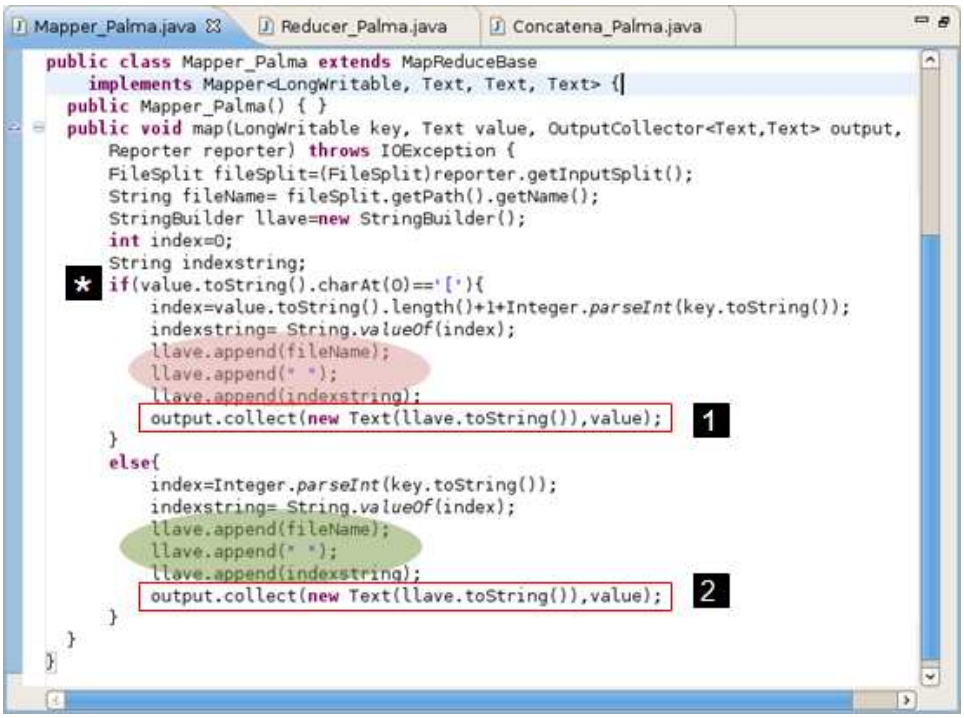
El paquete Palma_concatena especifica un proceso intermedio para Palma_accesos y Palma_recursos. Su función es acomodar los registros de los archivos de Palma.

Como se observa en la Figura 4-12, se utiliza una variable de tipo StringBuilder (llave) para almacenar el nombre del archivo de entrada concatenado a la suma del índice de la primera línea del registro con su longitud, formando la clave, los valores quedan intactos. Por esta razón tenemos dos opciones de salida que dependerán de que el registro contenga o no un carácter corchete. Si existe este carácter, se

produce como valor de salida la primera línea del registro y en caso contrario se toma la segunda línea.

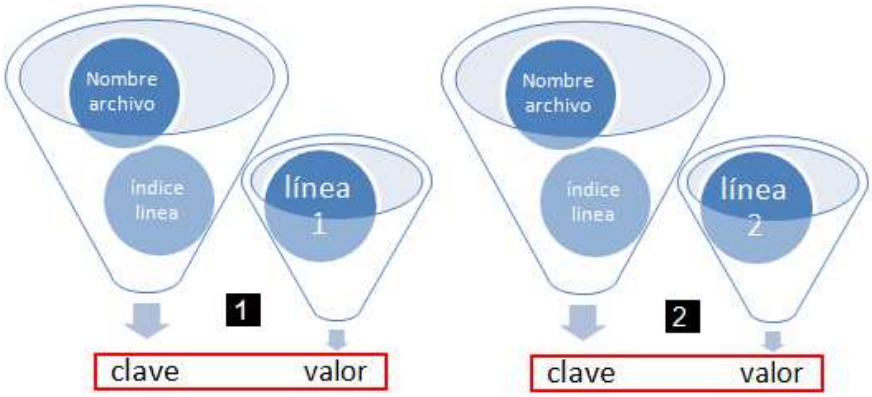
Se usaron colores que asocian y esquematizan cada una de las partes de la salida del mapper de Palma.

En la Figura 4-12 se muestra la manera de depurar los registros del log de Samba.



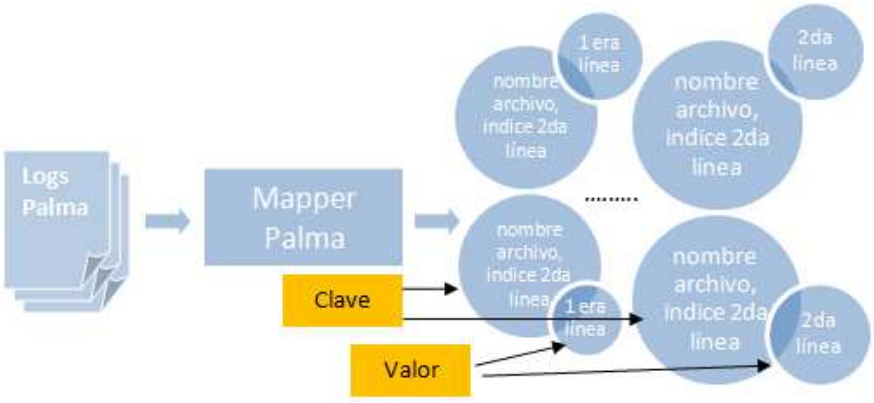
La Figura 4-12 Código Mapper de Palma

En la Figura 4-13 se grafica la salida del Mapper de Palma, representados con etiquetas que hacen referencia a la Figura 4-12.



La Figura 4-13 Código Mapper de Palma

En la Figura 4-14 se esquematiza los campos que contendrá cada registro mapeado de los archivos de Palma.



La Figura 4-14 Diagrama de Bloques del Mapper de Palma

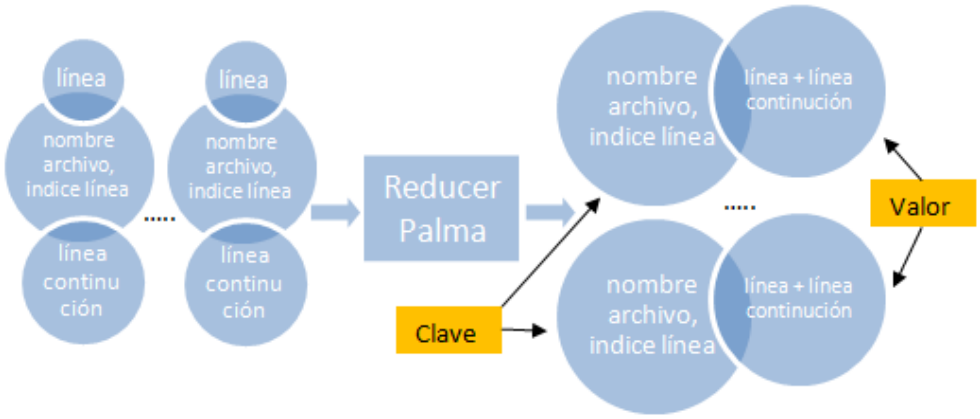
Con respecto al reducer de Palma se tiene lo siguiente:

En la Figura 4-15 se muestra como se concatenaron las líneas de un registro de Palma.

La Figura 4-15 Código Reducer de Palma

Se identificó el contenido de lo que se recibe en la variable registro; si es la primera o segunda línea, para posteriormente concatenar de manera ordenada.

El Figura 4-16 esquematiza los campos que se usaran en el reducer de Palma.



La Figura 4-16 Diagrama de Bloques del Reducer Palma

4.3 Presentación de los Resultados

Se utilizará Hive como motor de búsqueda para realizar consultas tipo SQL. La ventaja de este diseño es no repetir el proceso de extracción y manipulación de datos.

Para utilizar Hive, se usaron los archivos de texto plano producidos en la etapa de reducción. Inicialmente se crean tablas cuya estructura depende del formato del archivo. El contenido de los archivos de texto se carga en las tablas la carga de datos hacia la tabla y se procede a ejecutar las consultas.

El Figura 4-17 muestra el diagrama de presentación de los datos.

La Figura 4-17 Diagrama de Bloques del Hive

Para la presentación de los datos se tendrá una interfaz grafica que facilita al usuario la interacción con el sistema; haciendo transparente los procesos que se ejecutan a través de líneas de comandos.

Dentro de los parámetros de ejecución y consulta se mostrará un listado de las opciones disponibles que son:

1. HTTP
2. HTTPS
3. HFS
4. SSH
5. ICMP
6. DNS
7. IMAP
8. SMTP

El Figura 4-18 representa aplicaciones que se pueden escoger.

La Figura 4-18 Aplicaciones que se pueden elegir

Visitas a Cedro proporciona un listado de las IP desde las cuales se visita el servidor con más frecuencia.

La Figura 4-19 representa el detalle de presentación de las visitas de Cedro.

Detalles de Visitas a Cedro

Item	Fecha	IP	Protocolo	Frecuencia
1	02/Dec/2009	66.249.71.122	HTTP/1.1	46182
2	06/Dec/2009	66.249.65.175	HTTP/1.1	41716
3	07/Dec/2009	66.249.65.175	HTTP/1.1	40666
4	03/Dec/2009	66.249.71.122	HTTP/1.1	35327
5	08/Dec/2009	66.249.65.175	HTTP/1.1	33683
6	05/Dec/2009	66.249.65.175	HTTP/1.1	31994
7	04/Dec/2009	66.249.65.175	HTTP/1.1	19230
8	04/Dec/2009	66.249.71.122	HTTP/1.1	18357
9	09/Dec/2009	66.249.65.65	HTTP/1.1	16856
10	10/Dec/2009	66.249.65.65	HTTP/1.1	10441

Figura 4-19 Detalle de Presentación de Visitas de Cedro

Recursos de Cedro proporciona un listado de los recursos accedidos con más frecuencia.

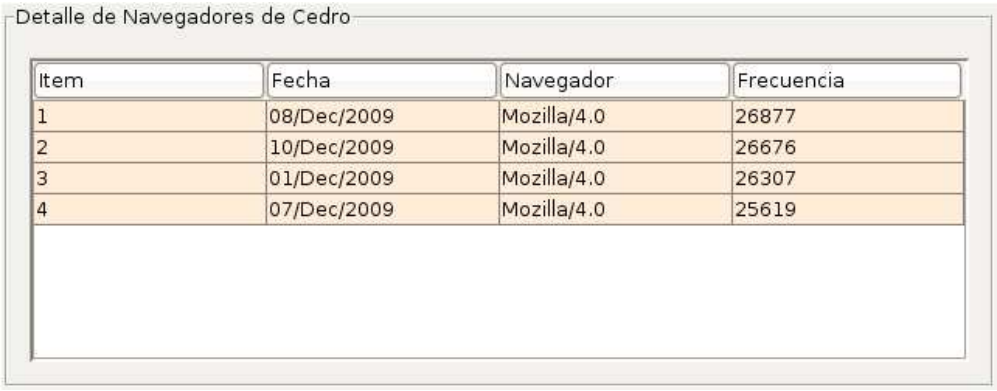
En la Figura 4-20 representa el detalle de presentación de recursos de Cedro. Observe que se muestra la frecuencia por recurso.


Detalle de Recursos de Cedro

Item	Fecha	Recurso	Frecuencia
1	07/Dec/2009	http://www.fiec.espol.edu.ec/templates/fiec_inicio_template/css/template_css.css	2375
2	08/Dec/2009	http://www.fiec.espol.edu.ec/index.php/Servicios/servicios.html	2214
3	02/Dec/2009	http://www.fiec.espol.edu.ec/index.php/Servicios/servicios.html	2175
4	10/Dec/2009	http://www.fiec.espol.edu.ec/templates/fiec_inicio_template/css/template_css.css	1960

Figura 4-20 Detalle de Presentación de los Recursos de Cedro

En la Figura 4-21 muestra un listado de los navegadores que se utilizan con más frecuencia.

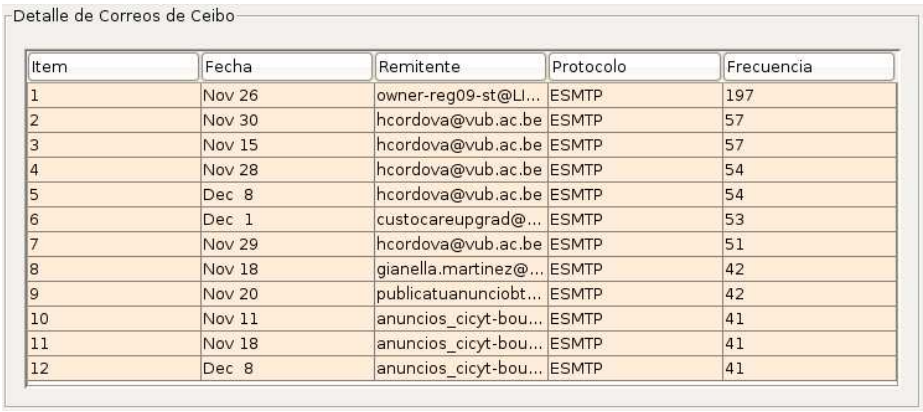

Detalle de Navegadores de Cedro

Item	Fecha	Navegador	Frecuencia
1	08/Dec/2009	Mozilla/4.0	26877
2	10/Dec/2009	Mozilla/4.0	26676
3	01/Dec/2009	Mozilla/4.0	26307
4	07/Dec/2009	Mozilla/4.0	25619

Figura 4-21 Detalle de Presentación de los Navegadores de Cedro

Correos de Ceibo proporciona un listado de los remitentes que envían correos con más frecuencia. También proporciona un listado de los destinatarios a los que le llegan correos con más frecuencia. El listado de remitentes o destinatarios se los obtiene escogiendo una de las opciones del filtro correos.

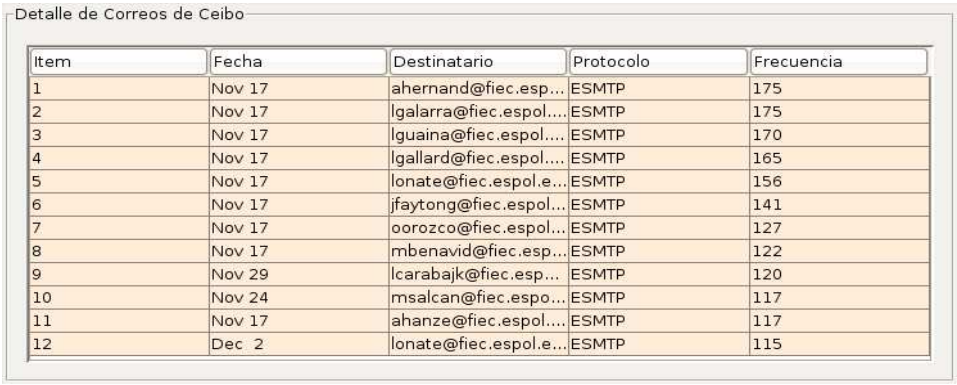
En la Figura 4-22 muestra como se detalla el resultado de ejecutar Correos de Ceibo (remitentes).


Detalle de Correos de Ceibo

Item	Fecha	Remitente	Protocolo	Frecuencia
1	Nov 26	owner-reg09-st@LI...	ESMTP	197
2	Nov 30	hcordova@vub.ac.be	ESMTP	57
3	Nov 15	hcordova@vub.ac.be	ESMTP	57
4	Nov 28	hcordova@vub.ac.be	ESMTP	54
5	Dec 8	hcordova@vub.ac.be	ESMTP	54
6	Dec 1	custocareupgrad@...	ESMTP	53
7	Nov 29	hcordova@vub.ac.be	ESMTP	51
8	Nov 18	gianella.martinez@...	ESMTP	42
9	Nov 20	publicatuanunciobt...	ESMTP	42
10	Nov 11	anuncios_cicyt-bou...	ESMTP	41
11	Nov 18	anuncios_cicyt-bou...	ESMTP	41
12	Dec 8	anuncios_cicyt-bou...	ESMTP	41

Figura 4-22 Detalle de Presentación de los Remitentes de Correos de Ceibo

En la Figura 4-23 muestra como se detalla el resultado al ejecutar Correos de Ceibo (destinatario).

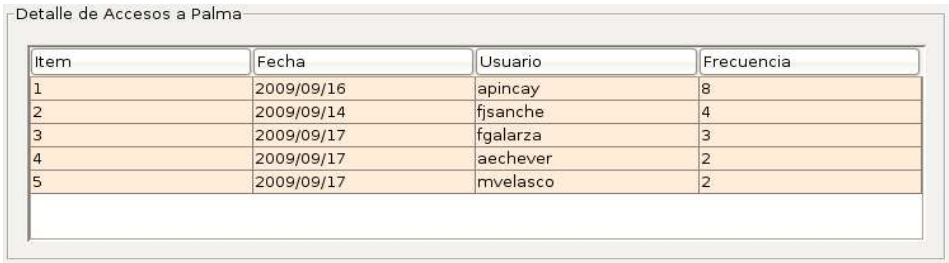

Detalle de Correos de Ceibo

Item	Fecha	Destinatario	Protocolo	Frecuencia
1	Nov 17	ahernand@fiec.esp...	ESMTP	175
2	Nov 17	lgalarra@fiec.espol....	ESMTP	175
3	Nov 17	lguaina@fiec.espol....	ESMTP	170
4	Nov 17	lgallard@fiec.espol....	ESMTP	165
5	Nov 17	lonate@fiec.espol.e...	ESMTP	156
6	Nov 17	jfaytong@fiec.espol...	ESMTP	141
7	Nov 17	oorozco@fiec.espol...	ESMTP	127
8	Nov 17	mbenavid@fiec.esp...	ESMTP	122
9	Nov 29	lcarabajk@fiec.esp...	ESMTP	120
10	Nov 24	msalcan@fiec.espo...	ESMTP	117
11	Nov 17	ahanze@fiec.espol....	ESMTP	117
12	Dec 2	lonate@fiec.espol.e...	ESMTP	115

Figura 4-23 Detalle de Presentación de los Destinatarios de Correos de Ceibo

Cuando se elije la opción Concatena_Palma se verifica si ha sido ejecutado previamente por el contrario se ejecuta este proceso antes de continuar.

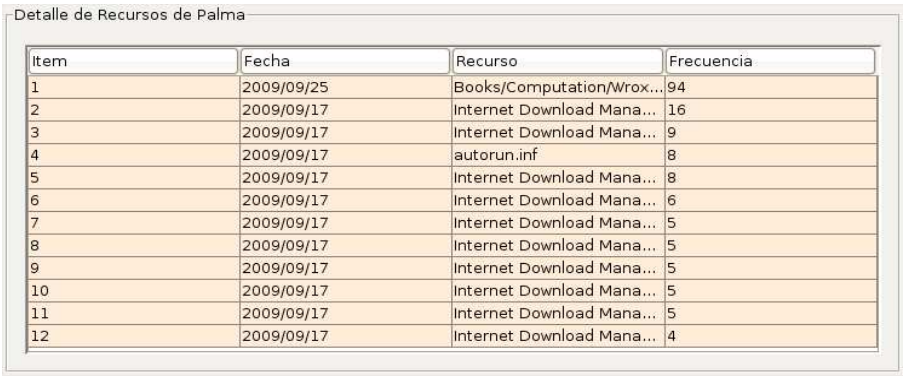
El Figura 4-24 representa detalle de los accesos de Palma.


Detalle de Accesos a Palma

Item	Fecha	Usuario	Frecuencia
1	2009/09/16	apincay	8
2	2009/09/14	fjsanche	4
3	2009/09/17	fgalarza	3
4	2009/09/17	aechever	2
5	2009/09/17	mvelasco	2

Figura 4-24 Detalle de los Accesos a Palma

En la Figura 4-25 se muestra el resultado de los Recursos de Palma proporcionando un listado de los recursos accedidos con mayor frecuencia, especificando la frecuencia por recurso


Detalle de Recursos de Palma

Item	Fecha	Recurso	Frecuencia
1	2009/09/25	Books/Computation/Wrox...	94
2	2009/09/17	Internet Download Mana...	16
3	2009/09/17	Internet Download Mana...	9
4	2009/09/17	autorun.inf	8
5	2009/09/17	Internet Download Mana...	8
6	2009/09/17	Internet Download Mana...	6
7	2009/09/17	Internet Download Mana...	5
8	2009/09/17	Internet Download Mana...	5
9	2009/09/17	Internet Download Mana...	5
10	2009/09/17	Internet Download Mana...	5
11	2009/09/17	Internet Download Mana...	5
12	2009/09/17	Internet Download Mana...	4

Figura 4-25 Detalle de los Recursos a Palma

CAPÍTULO 5

5. ANÁLISIS DE RESULTADOS

5.1 Metodología de las pruebas

Con el objetivo de evaluar la herramienta propuesta y desarrollada, se planificó un conjunto de pruebas que se describen en esta sección.

Para la ejecución del análisis de forma distribuida se usaron dos variables (número de nodos del clúster y tamaño total de la carga a procesar).

Se planteó realizar diez pruebas, donde se hace un muestreo de una variable (nodo), dejando la otra (carga total en gigas) como un valor fijo.

En cada prueba se procedió a levantar un clúster AWS con el número de nodos requerido (5-10-15), luego se anotaron los tiempos y se determinó la duración promedio por cada muestreo. Se calculó también la desviación estándar de los resultados tomados para cada prueba.

5.2 Resultados

Una vez recolectados los datos se procedió a realizar los gráficos e inferir resultados a partir de los mismos.

En la Tabla 5-1se muestran los resultados con 5-10-15 nodos.

		tiempo (min)								
		carga 37.2 MB / 0.036 GB								
		Palma Concatena			Palma Accesos			Palma Recursos		
		nodos								
		5	10	15	5	10	15	5	10	15
# prueba	1	6,783	1,933	1,367	0,467	0,350	0,267	0,417	0,620	0,333
	2	6,833	1,967	1,983	0,450	0,620	0,317	0,533	0,220	0,350
	3	7,017	3,020	1,683	0,517	0,200	0,300	0,550	0,240	0,350
	4	7,683	3,517	1,683	0,550	0,210	0,333	0,533	0,220	0,350
	5	6,833	4,350	1,267	0,517	0,270	0,300	0,433	0,600	0,300
	6	6,383	4,417	2,050	0,717	0,220	0,317	0,517	0,220	0,317
	7	9,683	5,667	1,633	0,483	0,210	0,300	0,450	0,630	0,400
	8	6,167	4,317	1,483	0,467	0,600	0,283	0,450	0,240	0,300
	9	8,633	4,267	1,283	0,433	0,590	0,417	0,500	0,680	0,317
	10	6,350	5,367	1,333	0,517	0,230	0,317	0,517	0,200	0,317
	promedio	**7,237**	**3,882**	**1,577**	**0,512**	**0,350**	**0,315**	**0,490**	**0,387**	**0,333**
	Var. Std.	**1,124**	**1,273**	**0,281**	**0,081**	**0,180**	**0,040**	**0,048**	**0,213**	**0,030**

Tabla 5-1 Pruebas Palma con número de nodos variable. Carga de 0.036 GB

Con los valores obtenidos en la Tabla 5-1 se generó la Figura 5-1.

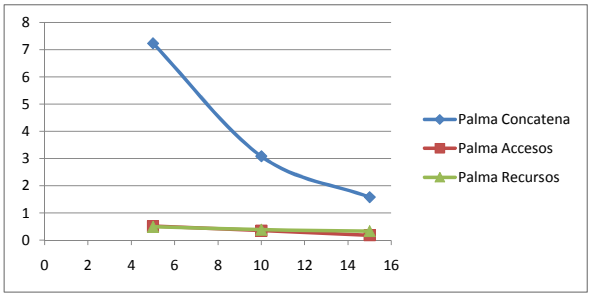


Figura 5-1 Resultados promedio de pruebas Palma con número de nodos variable (nodos VS tiempo (min)).

De la Figura 5-1, se puede observar que el proceso de Palma Concatena consume más recursos, debido a que el formato de la data es más complejo. La mejora es notoria cuando se duplican los nodos en ambos tramos (de 5 a 10 y de 10 a 15); mejorando un 46,35% y 59,38% en tiempo de procesamiento respectivamente.

Se puede mencionar que en todos los procesos de 10 nodos, se ve que la desviación estándar es mayor, lo cual significa que con un clúster de 10 nodos se tienen las muestras más dispersas en promedio de la media.

Se realizó pruebas con 5, 10 y 15 nodos, y los resultados (en minutos) se muestran en la Tabla 5-2.

		tiempo (min)								
		793 MB / 0.774 GB								
		Cedro Navegadores			**Cedro Visitas**			**Cedro Recursos**		
		Nodos								
		5	10	15	5	10	15	5	10	15
# prueba	1	0,517	0,417	0,417	0,517	0,517	0,467	0,367	0,300	0,300
	2	0,467	0,400	0,400	0,550	0,500	0,483	0,367	0,283	0,283
	3	0,467	0,350	0,383	0,567	0,450	0,483	0,317	0,300	0,267
	4	0,483	0,433	0,400	0,533	0,483	0,483	0,350	0,267	0,267
	5	0,467	0,400	0,383	0,550	0,467	0,450	0,367	0,283	0,283
	6	0,450	0,400	0,400	0,500	0,483	0,467	0,333	0,367	0,290
	7	0,483	0,400	0,400	0,617	0,533	0,467	0,300	0,300	0,300
	8	0,483	0,417	0,383	0,617	0,533	0,450	0,300	0,267	0,267
	9	0,467	0,417	0,417	0,517	0,417	0,450	0,333	0,283	0,283
	10	0,517	0,417	0,383	0,550	0,517	0,483	0,383	0,283	0,283
	promedio	**0,480**	**0,405**	**0,397**	**0,552**	**0,490**	**0,468**	**0,342**	**0,293**	**0,282**
	Var. Std.	**0,022**	**0,022**	**0,013**	**0,040**	**0,038**	**0,015**	**0,030**	**0,029**	**0,013**

Tabla 5-2 Pruebas Cedro con número de nodos variable. Carga de 0.774 GB

Con los valores obtenidos se generó la Figura 5-2.

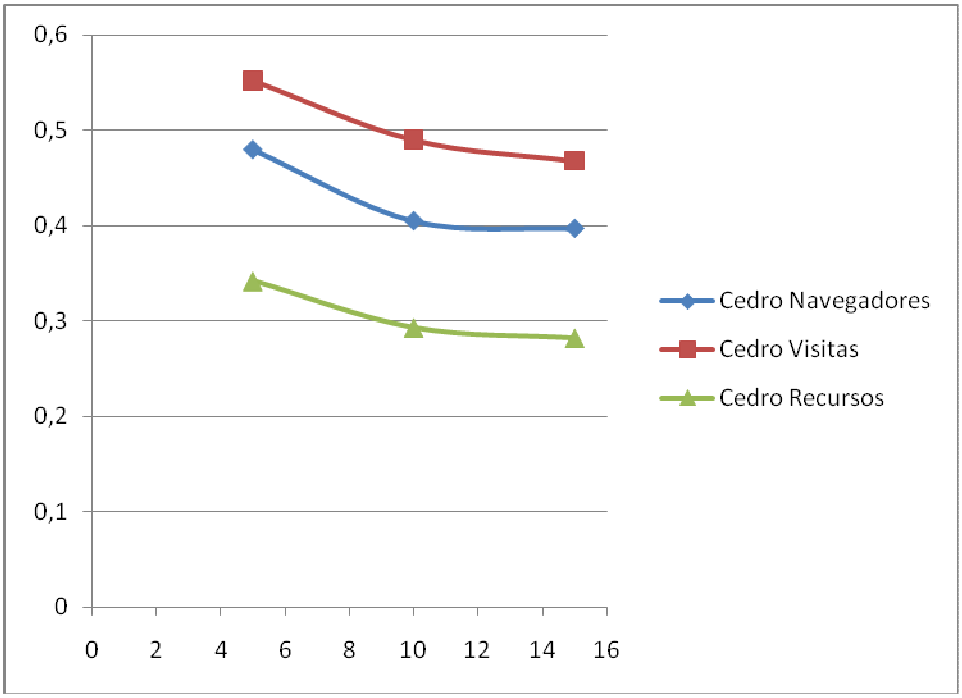


Figura 5-2 Resultados promedio de pruebas Cedro con número de nodos variable (nodos VS tiempo (min)).

De la Figura 5-2, se puede observar que todos los procesos presentan tiempos similares desfasados aproximadamente 0.1 min; es decir que Cedro Recursos es el más rápido, a continuación Cedro Navegadores y finalmente Cedro Visitas.

El tiempo de procesamiento mejora en un 13,73% cuando la carga es de 0,774 GB en el intervalo de 5 a 10 nodos; mientras que en el tramo de 10 a 15 nodos el promedio mejora en 3,75%.

Se observa también de ésta prueba que la desviación estándar de los resultados disminuye a medida que se incrementa el número de nodos, lo que indica que se puede generar cálculos de tiempos estimados de ejecución más confiables con un mayor número de nodos en el clúster.

Se realizó pruebas con 5, 10 y 15 nodos, y los resultados (en minutos) se muestran en la Tabla 5-3.

		tiempo (min)		
		562 MB / 0.549 GB		
		Ceibo Correos		
		nodos		
		5	10	15
# prueba	1	0,493	0,383	0,317
	2	0,467	0,367	0,367
	3	0,433	0,350	0,400
	4	0,450	0,400	0,383
	5	0,467	0,400	0,350
	6	0,467	0,383	0,317
	7	0,450	0,383	0,350
	8	0,417	0,360	0,350
	9	0,450	0,383	0,317
	10	0,417	0,400	0,367
	promedio	**0,451**	**0,381**	**0,352**
	Var. Std.	**0,024**	**0,017**	**0,029**

Tabla 5-3 Pruebas Ceibo con número de nodos variable. Carga de 0.549 GB

Con los valores obtenidos de la Tabla 5-3 se generó la Figura 5-3.

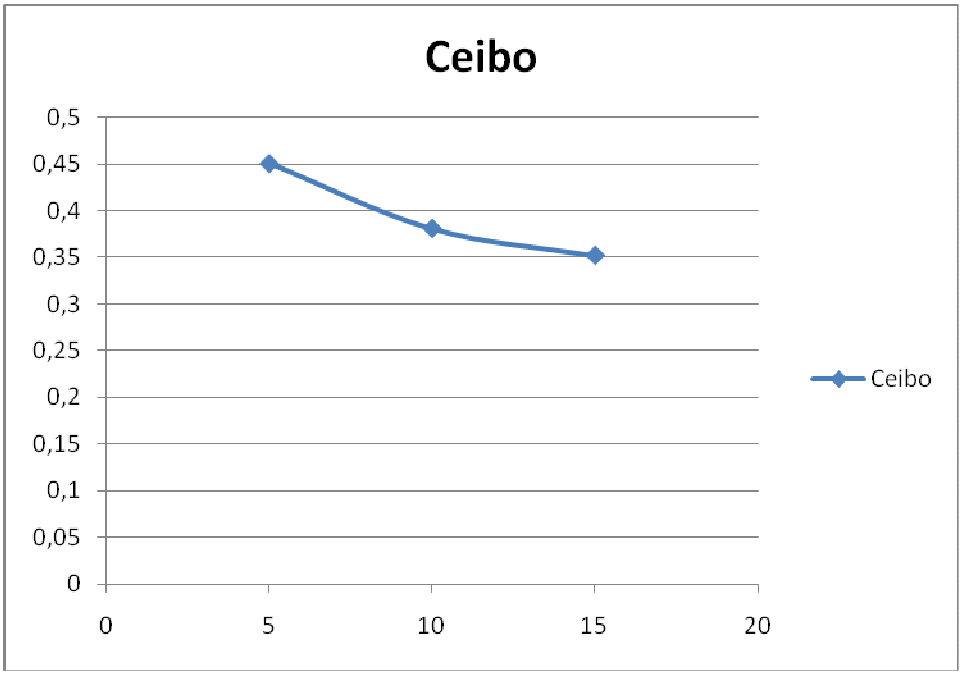


Figura 5-3 Resultados promedio de pruebas Ceibo con número de nodos variable (nodos VS tiempo (min)).

De la figura, se puede observar que con 10 nodos parece ser un número adecuado de nodos para un buen rendimiento sin necesidad de un clúster de mas capacidad, aquí serán imperceptibles los progresos en mejora de tiempo.

El promedio del procesos con 0,549 GB cuando se duplican los nodos en el tramo de 5 a 10; es 15,52%.

5.3 Comparación de tiempos con herramientas existentes

Con el fin de medir el comportamiento obtenido con la herramienta frente a soluciones no distribuidas de logs, se realizó una prueba con M5 Analyzer, con una carga de 10GB, utilizando un core por cada procesador obteniendo los siguientes resultados:

La Figura 5-4 muestra el resultado de ejecutar M5 Analyzer con 10GB.

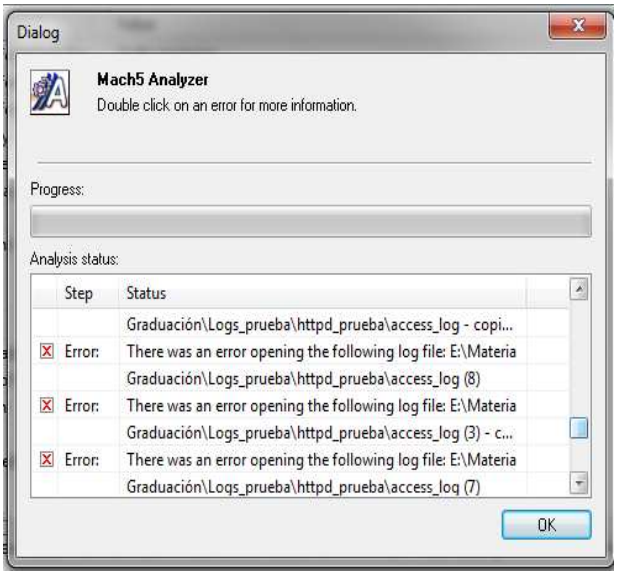


Figura 5-4 Resultados M5 Analyzer.

La Figura 5-5 muestra el resultado de ejecutar M5 Analyzer con 10GB.

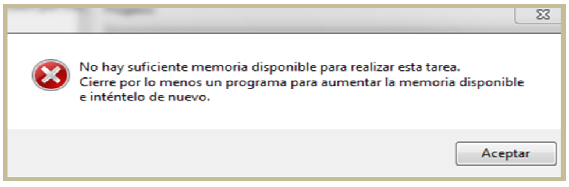


Figura 5-5 Resultados M5 Analyzer.

Lo cual indica que la herramienta no es escalable, en contraparte con la herramienta diseñada en Hadoop que soporta más carga dependiendo del espacio en disco del duro que se tenga.

En una de las pruebas realizadas utilizando la herramienta con la supervisión del administrador de redes de FIEC, se determinó un posible ataque que fue tomado en cuenta por la extremada frecuencia de visitas en un solo día al servidor de Cedro.

La Figura 5-6 muestra que un rango de IPs con las frecuencias para las visitas a Cedro, las frecuencias oscilan en valores promedio de 45000 visitas por día.

IP	Protocolo	Frecuencia
66.249.71.122	HTTP/1.1	46182
66.249.65.175	HTTP/1.1	41716
66.249.65.175	HTTP/1.1	40666
66.249.71.122	HTTP/1.1	35327
66.249.65.175	HTTP/1.1	33683

Figura 5-6 Resultados M5 Analyzer.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Se determino que para estabilizar el procesamiento distribuido con una carga de aproximadamente un 1GB se requieren entre 5 y 10 como el número de nodos adecuado para un buen rendimiento sin necesidad de un clúster de mayor capacidad, aquí serán imperceptibles los progresos en mejora de tiempo.
2. El análisis de logs es un proceso que puede ser extenso cuando se procesa secuencialmente archivos de gran tamaño.
3. Aplicar un modelo distribuido para el análisis de logs genera soluciones altamente escalables, y la tolerancia a fallos puede ser controlada.
4. La extracción de los campos apropiados para luego presentarlos de manera condensada permiten detectar anomalías en accesos a servidores de forma más eficaz.
5. La herramienta implementada que soporta procesamiento distribuido supera a las demás en rendimiento, ya que no brindan soporte para manipulación de archivos de mayor tamaño.

RECOMENDACIONES

1. Se recomienda previa a la instalación de hadoop, asignar suficiente espacio en disco para poder procesar información que crece con el tiempo de manera rápida.

2. Los administradores de red podrán adaptar esta propuesta que se ha detallado para presentar resultados de consultas personalizadas a sus necesidades, pero para eso tendrían que familiarizarse con el entorno que ofrece Hadoop y su aplicación Hive.

BIBLIOGRAFÍA

[1] Netcraft. *"August 2010 Web Server Survey".* [Online] [Cited: Agosto 11, 2010.] http://news.netcraft.com/archives/2010/08/11/august-2010-web-server-survey-4.html#more-2752.

[2] **Mateo Carles.** *"Desarrollo de Aplicaciones web".* FUOC XP06/M2108/01497. Universidad Oberta de Catalunya, Marzo 2004. pag. 232-234.

[3] **Areitio Bertolín Javier, Areitio Bertolín Ana.** *"Test de penetración y gestión de vulnerabilidades, estrategia clave para evaluar la seguridad de red",* REE. España, Abril 2009: pag. 40-42.

[4] **Dean Jeffrey, Ghemawat Sanjay "***Mapreduce: Simplified Data Processing on Large Clusters*". Presentadoen el USENIX Symposium on Operating Systems Design & Implementation. (OSDI) 2004: Cap. 2 pag. 2-3.

[5] **Venner Jason** *"Pro Hadoop: Build scalable, distributed aplications in the cloud".* Apress, EE-UU 2009: Cap.1 pag.1-24.

[6] **Dean Jeffrey, Ghemawat Sanjay** *"MapReduce: Simplified Data Processing on Large Clusters"*. Google's Mountain View, CA-EEUU. Enero 2008/Vol. 51, No. 1. Cap. 3.3 pag. 109.

[7] **Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, Raghotham Murthy.**

“Hive - A Warehousing Solution Over a Map-Reduce Framework”. Facebook Data Infrastructure Team, cap. 3.

[8] Amazon Web Services. *“Amazon Simple Storage Service (Amazon S3) ”.* [Online] [Cited: 2010.] http://aws.amazon.com/s3/.

[9] Amazon Web Services. *“Amazon Elastic Compute Cloud (Amazon EC2)”.* [Online] [Cited: 2010.] httphttp://aws.amazon.com/ec2/.